5.10 Exercises

1. Write a program, in which you:

   a) Define two pointers pv1 and pv2, where pv1 can point to a one dimensional real array and pv2 can point to a two dimensional real array.

   b) Define two target real arrays, tv1 and tv2, where tv1 is one dimensional with bounds -3:5 and tv2 is two dimensional with bounds 1:5, 1:10.

   c) Set up the array pointer pv1 to point to tv1 such that pv1 has the lower bound -3 (Write out the lower bound of pv1 for confirmation).

   d) Set up the array pointer pv1 to point to tv1 such that pv1 has the lower bound 1 (Write out the lower bound of pv1 for confirmation).

   e) Can you set pv1 to point to tv1 such that pv1 has the lower bound -2?

   f) Use pointers (pv1 or pv2) to write out the 4th row of tv2, the section tv2(2:4, 4:8) and the section tv1(1:5:2).

   (`p_array.f90`)

2. Look at the program `status.f90`, write down what you think will be printed. Then run the program to compare.

3. Write a program which uses an array of pointers (simulated by means of a derived type having a pointer component of the desired type) to set up a lower-triangular matrix. (`p_matrix.f90`)

4. Run the program `simple.f90`, notice that the linked list stores the typed-in numbers in reverse order. Modify this program such that the linked list preserves the order of typed-in numbers. (`linklist.f90`)

5. Run the program `polyline.f90`, which uses the polyline module (`poly_mod.f90`), notice that the linked list stores the points read in reverse order. When prompted for a y value enter the y value of the point that you want to delete from the list. Compare the two versions of the list that have been printed. Has your point been deleted?

   Modify this program such that the linked list preserves the order of points read. (`poly2.f90`)